Self-Adaptive Architecture for Multi-sensor Embedded Vision System

Ali Isavudeen, Eva Dokladalova, Nicolas Ngan, Mohamed Akil

Self-Adaptive Architecture for Multi-sensor Embedded Vision System

Ali Isavudeen[1,2], Eva Dokladalova[1], Nicolas Ngan[2], and Mohamed Akil[1]

[1] Laboratoire Informatique Gaspard Monge, Equipe A3SI
Unit Mixte CNRS-UMLV-ESIEE (UMR 8049)
Noisy-le-Grand, France
{eva.dokladalova, akil.mohamed}@esiee.fr
[2] Sagem Défense et Sécurité
Groupe Safran
Argenteuil, France
{ali.isavudeen, nicolas.ngan}@sagem.com

Abstract. Architectural optimization for heterogeneous multi-sensor processing is a real technological challenge. Most of the vision systems involve only one single color sensor and they do not address the heterogeneous sensors challenge. However, more and more applications require other types of sensor in addition, such as infrared or low-light sensor, so that the vision system could face various luminosity conditions. These heterogeneous sensors could differ in the spectral band, the resolution or even the frame rate. Such sensor variety needs huge computing performance, but embedded systems have stringent area and power constraints. Reconfigurable architecture makes possible flexible computing while respecting the latter constraints. Many reconfigurable architectures for vision application have been proposed in the past. Yet, few of them propose a real dynamic adaptation capability to manage sensor heterogeneity. In this paper, a self-adaptive architecture is proposed to deal with heterogeneous sensors dynamically. This architecture supports on-the-fly sensor switch. Architecture of the system is self-adapted thanks to a system monitor and an adaptation controller. A stream header concept is used to convey sensor information to the self-adaptive architecture. The proposed architecture was implemented in Altera Cyclone V FPGA. In this implementation, adaptation of the architecture consists in Dynamic and Partial Reconfiguration of FPGA. The self-adaptive ability of the architecture has been proved with low resource overhead and an average global adaptation time of 75 ms.

1 Introduction

Performance capability of modern embedded vision system is increasing day by day. Requirements of a vision system mostly depend on the application of the system, and priorities on performance criteria will be different whether it is for public, automotive, medical or military purpose. In military application for instance, vision systems have to face multiple and various environmental and

operational contexts. The system should face all luminosity conditions, be it day, night, indoor or outdoor environment. Operational context could either be surveillance, tracking or targeting.

Most of time, both industrial and academic architecture of embedded vision system integrate only one single image sensor. The widely used and known one is the CMOS color sensor. This sensor can capture and produce a color digital picture[1]. However, the CMOS color sensor is not sufficient to face the variable environmental and luminosity context. The vision system has to integrate different types of sensor such as color, infrared or intensified light sensor. These sensors differ in many characteristics such as the spectral band, the Field of View (FOV), the frame resolution, the frame rate or even the pixel size. This sensor variety makes difficult the architecture designing. Actually, in the case of heterogeneous sensors vision system, the processing architecture has to carry out different type of data at different frequency. While general purpose processor-based systems are often designed for uniform data processing, reconfigurable computing [2] offers the possibility to carry out heterogeneous data processing.

Many reconfigurable hardware based or reprogrammable software based solutions have been proposed previously [3][4][5]. Nevertheless, these architectures do not support dynamic replacement of the sensor. The real challenge that we want to tackle here is the dynamic adaptability of the architecture in a heterogeneous multi-sensor vision system. The expected system should be able to adapt its processing architecture dynamically to manage on-the-fly replacement of the sensor.

In this paper, we propose a novel self-adaptive architecture for heterogeneous sensors vision system. The architecture can self-adapt its hardware organization in response to a dynamic switch of the sensor. A stream header conveying information on the sensor is included in the image stream. This information is used to adapt the processing architecture according to the sensor characteristics.

The paper is organized as follows. An overview of related works is given in section 2. The new self-adaptive architecture and its features are presented in section 3 while the experimental prototype for evaluation is presented in section 4. Experimental results are discussed in section 5. Finally, section 6 concludes and announces perspectives for future work.

2 Multi-sensor Embedded Vision System

Photo and video camera are the most widely known vision systems. In these systems, as like as in most of the vision systems, either academic or industrial one, there is only one CMOS color sensor. The CMOS sensor is good enough for day vision purpose but it is not sufficient for variable luminosity condition. Several other sensors, such as infrared or low-light sensor, are used for medical, automotive or military applications. Infrared sensors, also known as thermal sensors, are sensible to the infrared waves which stem from heat objects. The specificity of the low-light sensors is their ability to work in very low luminosity condition.

Multi-sensor concept has also been used in stereo vision systems [6] [7]. In these vision systems there are two image sensors, but both of them have same characteristics. They also have the same image processing for both sensors and the processing for both sensor is done in parallel. Our aim is to propose an adaptive processing architecture for vision system involving different kind of sensor.

Processing architecture in many embedded vision system is based on a software solution with a DSP or general purpose processor. Most of the embedded vision systems use an ARM-based video processing core, with an additional DSP and specific image&video co-processing cores [8]. These solutions are mostly designed for a given sensor, which is often a color sensor. Their poor flexibility performance is not suitable for multiple heterogeneous sensor vision systems.

Reconfigurable hardware, such as Field Programmable Gate Arrays (FPGA), are good compromise for high performance, high flexibility and reasonable power consumption. Many works have explored FPGA-based reconfigurable architecture for vision application [3][4][5]. In [4], Dynamic and Partial Reconfiguration (DPR) is used to evaluate acceleration performance of some image processing in FPGA implementation. In [5], authors explore DPR to implement an automatic white balance algorithm. Both works do not deal with heterogeneous sensors.

The DreamCam proposed in [3] presents an FPGA implementation for Harris&Stephens corner and edge detection algorithm. This work proposes a reconfigurable solution for embedded cameras involving heterogeneous sensors. However, the architecture can be only statically reconfigured when the sensor is switched. For a given image processing chain, the architecture enables only parameter modifications. The sensor can not be switched dynamically.

Our work attempts to provide a self-adaptive architecture that supports vision system involving multiple and heterogeneous sensors. This architecture can dynamically adapts its organization to enable dynamic switch of the sensor.

3 Self-Adaptive Multi-sensor system

A multi-sensor vision system involves more than one image sensor. There are many types of sensor that a vision system could integrate. The Table 1 gives some examples of existing sensors and their characteristics.

Type	*Spectral band*	*Color space*	*Resolution*	*Frame rate (fps)*	*Pixel size (bit)*
Color [9]	visible	RGB	1280x960	45	3x12
Low-light [10]	visible	Grayscale	1280x1024	60	10
Infrared [11]	infrared	Grayscale	640x480	120	12

Table 1. Example of sensors characteristics

The proposed self-adaptive architecture is supposed to manage such a variety of sensors mentioned in the Table 1. In our framework, we suppose that only one sensor is active at a time while the remaining sensor are unused. The global architecture of the vision system is presented in the Fig. 1.

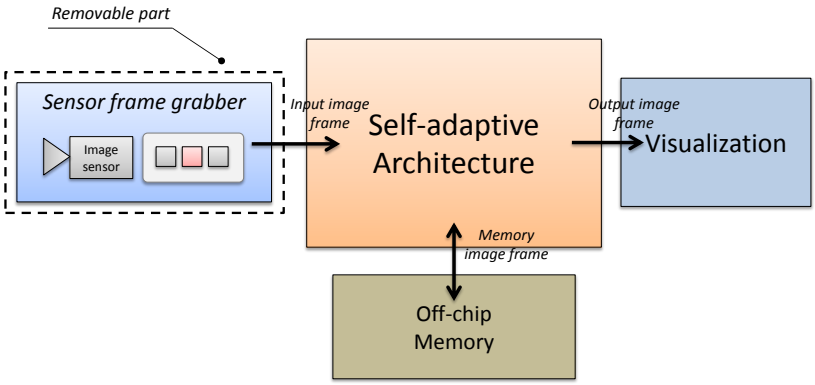

Fig. 1. Architecture of the vision system

The system is mainly composed of a frame grabber including the sensor, the proposed Self-adaptive architecture, visualization hardware and external computing resources such as off-chip memory. The Sensor frame grabber is a removable part that can be replaced when the sensor has to be changed.

3.1 Sensor frame grabber

Whatever is the sensor type, each Sensor frame grabber has the same hardware organization. The Fig. 2 depicts the sensor frame grabber organization.

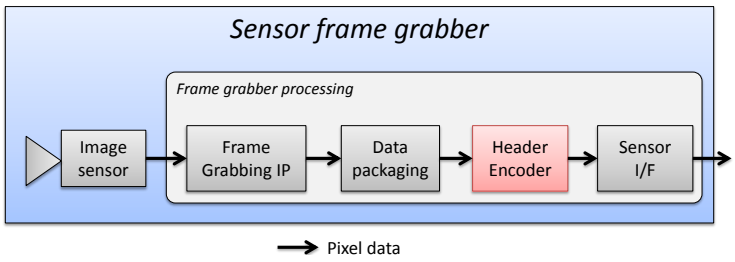

Fig. 2. Architecture of the sensor preprocessing

The image sensor is followed by the Frame Grabbing Intellectual Property (IP). This IP is specific to the image sensor and it is used to drive the latter and readout the pixel values from the imagers pixel array. Then the image frame data are gathered into packets and sent to the processing architecture by the communication core. The novelty that this work brings into the Sensor frame grabber is the Header Encoder.

This Header Encoder adds a Stream Header to the existing image stream. This Stream Header is used as a identity card of the sensor. It conveys the sensor's characteristics to the Self-adaptive architecture.

3.2 Stream Header

Packet header concept is widely used in Network-On-Chip to add a description of the packet within the packet. In real time vision application, this concept was used to add description of the image frame into the image packets [12]. In our system, the Header Encoder is placed next to the Data packaging processing in the Sensor frame grabber. The Stream Header does not need any extra datapath. It is integrated into the image stream, so that the image data and the Stream Header data use the same datapath.

The Stream Header contains information that intends to be used to adapt the image processing in the Self-adaptive architecture. The Stream Header is mainly composed of the sensor's characteristics. Its details are given in the Fig. 3. In this figure, frame synchronizations are presented in blue arrow. They announces the beginning of a frame. The Stream Header in inserted between the frame synchronization and the image frame data.

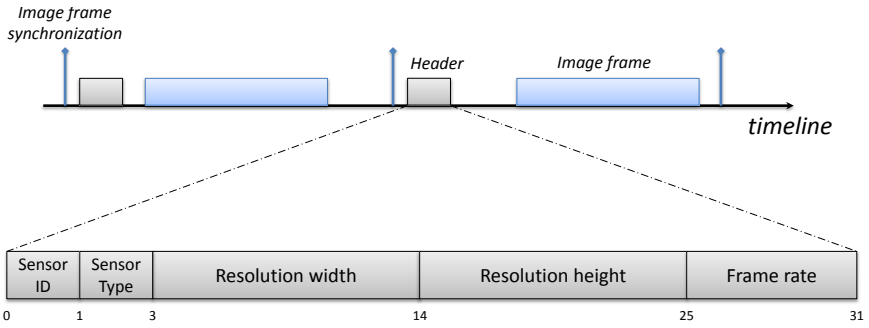

Fig. 3. Stream Header data format

The Stream Header is composed of five data. There are the sensor ID, the sensor type, the resolution separated into resolution width and resolution height and finally the frame rate. The sensor ID has fulfils two functions. In one hand, it is used to recognize the given sensor among several sensors that the vision system is ought to deal with. In other hand, the sensor ID will be used by the

Self-adaptive architecture to detect when the sensor is switched. The sensor type is the main information that is used to adapt the image processing.

3.3 Self-adaptive Architecture

Image frame data and Stream Header are then processed in the Self-adaptive architecture. The Fig. 4 depicts the Self-adaptive architecture and its components. Input image stream coming from the Sensor frame grabber is collected through the Sensor interface (I/F), then processed in the processing area, and finally sent to the visualization hardware through the Visualization interface. Frame buffers are used between the processing area and the external communication interface both sensor and visualization sides, in order to separate the pixel clock frequencies.

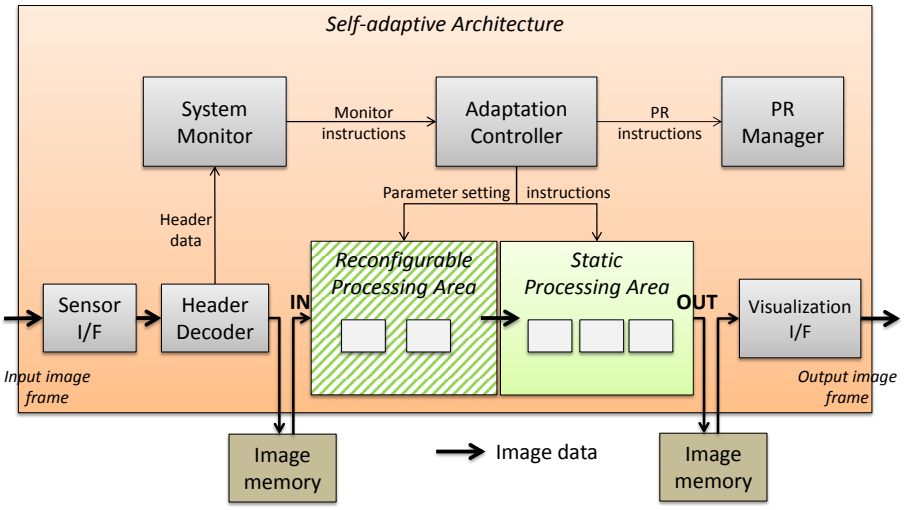

Fig. 4. Self-adaptive architecture

The adaptation brain of the Self-adaptive architecture is composed of four main components : Header Decoder, System Monitor, Adaptation Controller and Partial Reconfiguration Manager (PR Manager).

The input image stream is first introduced in the Header Decoder. The Header Decoder extracts data from the Stream Header. The extracted data is then sent separately to the System Monitor. The output image frame from the Header Decoder does not contain the Stream Header anymore, only the image data goes through the processing area.

All the image processing are implemented in the processing area. This area is divided into two parts : Reconfigurable processing area and Static processing area. Actually, the image processing chain to process the sensor data can be

divided into two parts. The first part of the processing depends on the sensor type whereas the second one is common for every sensor. Image processing chain of two different sensors differs only in the first part. Details on the considered image processing chains in this work are given in the case of study in section 4.

When the sensor is switched, only the first part of the image processing chain is modified. Hence, this part is placed in the Reconfigurable processing area. The adaptation of the first part consist in Dynamic and Partial Reconfiguration (DPR) of the Reconfigurable processing area. In the second part, only resolution parameters are adapted without modifying the rest of image processing chain.

3.4 Adaptation process

The System Monitor collects Stream Header data coming from the Header Decoder. It has local registers where these data can be saved. The System Monitor first compares the previously saved sensor ID and the new sensor ID. If the two IDs are not the same, it turns a *new-sensor* flag to active state to inform the Adaptation Controller that the sensor has been switched. Then it saves the new Stream Header data in the local registers.

The System Monitor has also an additional register to save the current sensor type of the system. This information will be used by the Adaptation Controller. Finally, the System Monitor transmits the saved data to the Adaptation Controller.

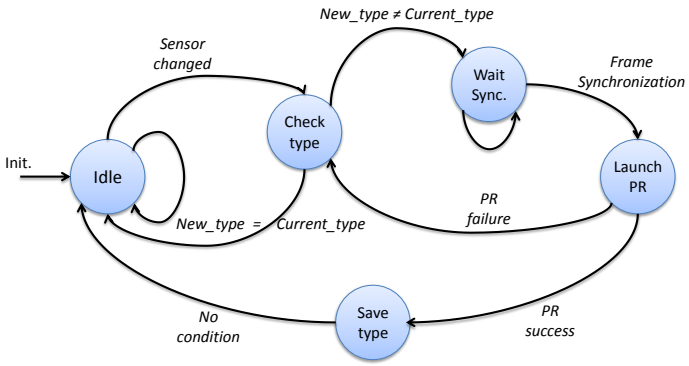

Fig. 5. Finite State Machine of the adaptation process

The Adaptation Controller is responsible for the adaptation of the image processing when the sensor is switched. The adaptation process follows the Finite State Machine (FSM) presented in the Fig. 5.

This FSM has five states. Initially, the system starts with the default **Idle** state. If the *new-sensor* flag is active, it means that the sensor has been switched. Consequently, the FSM goes to the **Check type** state. In this state, the current

sensor's type is compared to the new sensor's type. If the new one is different than the current one, then the **Wait sync** state is reached. Otherwise, the FSM returns to **Idle** state.

In the **Wait sync** state the FSM waits for the next image frame synchronization before sending a PR request. Actually, if the Partial Reconfiguration is launched in the middle of the processing of a frame, it will corrupt the output image stream. Once a image frame synchronization has passed, the Adaptation Controller fires PR request to the PR Manager (**Launch PR**). The Adaptation Controller also gives the new sensor's type so that the PR Manager configures the right image processing chain in the Reconfigurable processing area.

When the Partial Reconfiguration has finished successfully, the System Monitor saves the new sensor's type (**Save status**) before coming back to the **Idle** state. In case of a PR failure, the FSM returns to **Check type** state in order to restart the adaptation process.

4 Experimental prototyping

Evaluation of the proposed Self-Adaptive architecture concept has been made in a case of study with two sensors. A test bench with one color sensor and one infrared sensor has been used for experimental purpose only. The Fig. 6 depicts the Self-adaptive architecture with this test bench. In this test bench, the selection of the sensor is made by a manual switch. This test bench intends to make easy the technical manipulations to switch between the two sensors, otherwise the concept of the Self-adaptive architecture remains unchanged.

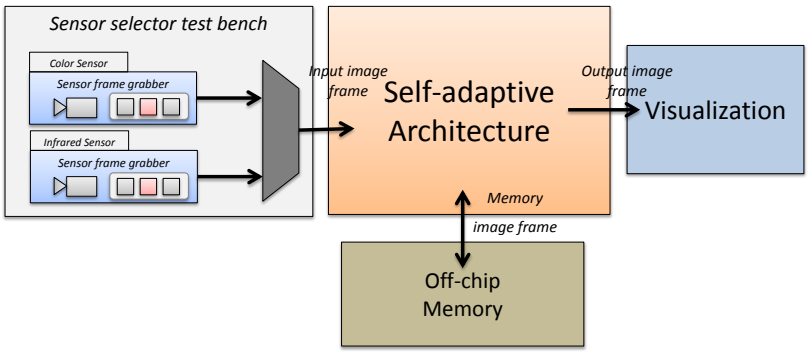

Fig. 6. Architecture of the vision system with the test bench

Image processing chain for color and infrared sensor of this case of study are given in Fig. 7. The Reconfigurable Area contains the image processing that are specific to the sensor. These processing perform image restoration.

In the case of the color sensor, these processing are White Balance, Debayer also know as Demosaicing and finally color space transform processing to

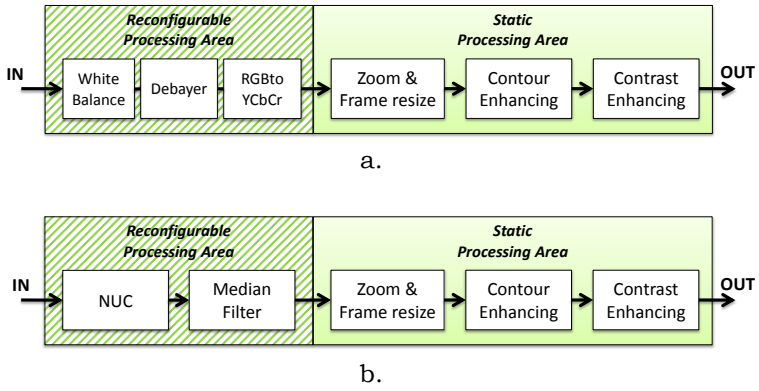

Fig. 7. Image processing chain for color(a) and infrared(b) sensor

switch from RGB space to YCbCr space. In the case of the infrared sensor, the restoration processing are Non-Uniformity-Correction (NUC) and Median Filter processing.

Image quality enhancement processing are common for both color and infrared sensor so they are placed in the static area. These processing are Digital zoom and frame size readjusting, Contour Enhancing and Contrast Enhancing.

5 Evaluation and Results

The proposed architecture was implemented in a Cyclone V Altera FPGA [13]. In FPGA implementation, area occupation of an architecture is given in terms of logical and memory resources, which are look-up-tables(LUT), registers and memory blocks.

	ALUT	Register	Memory (bit)
Reconfigurable Area	12 240	24 480	1 044 480
Reconfigurable Color processing	1 733 (14%)	1 678 (6.9%)	122 800 (12%)
Reconfigurable Infrared processing	1 747 (14%)	5 055 (21%)	98 304 (9.4%)

Table 2. Resource utilization of processing chain

By implementing on a recent FPGA, this work highlights the advantage of the partial reconfiguration for resource optimization. Instead of using two static

processing areas for the changing part of the image processing chain, resources of the same Reconfigurable processing area are temporally multiplexed for both color and infrared sensor. The size of Reconfigurable processing area depends on the resource requirements of the design that is deployed in the Reconfigurable processing area. This area shall include all the resources required by anyone of the two reconfigurable processing chains. Usually, the reconfigurable area includes little bit more resources than the exact required amount.

The Table 2 gives details about the selected reconfigurable area in the proposed prototype. It gives available resources in the reconfigurable area. Percentage between brackets in Table 2 represents the fraction of resources used by each reconfigurable processing chain within the total resources available in the reconfigurable area. The selected reconfigurable area has a 16-columns width and a 51-rows height.

We can see in the Table 2 that the amount of resources in the reconfigurable area is extremely higher than the resources required from anyone of the two reconfigurable processing chains. This unusual extra resources in the reconfigurable area is due to the limitation of the Partial Reconfiguration flow in Altera FPGA. A smaller area than the proposed one in the Table 2 leads to a compilation failure.

In Xilinx Partial Reconfiguration flow, the reconfigurable area can be finely defined, so that we get small amount of unused resources. In [5], authors relate that only 25 % of unused resources in the PR area is enough to avoid compilation failure.

	ALUT	Register	Memory (bit)
Full Design	24 947	31 945	1 726 056
Header Encoder	43	5	0
Header Decoder	12	24	0
System Monitor	8	28	0
Adaptation Controller	31	22	0
PR Manager	22	15	0
Total Adaptation Components	**116 (0,5 %)**	**94 (0,3 %)**	**0**

Table 3. Resource overhead of adaptation components

Resource overhead of the adaptation components has also been evaluated and reported in the Table 3. We can see in these results that the total resource overhead of the adaptation components are quite insignificant compared to the resource utilization of the full design. This low resource overhead is justified by the modest complexity of the proposed adaptation process.

Adaptation times are given in the Table 4. This table gives adaptation time in milliseconds for a system clock of 100 MHz. Most of the adaptation time is due to the partial reconfiguration of the FPGA. An average adaptation time of

	Time (ms)
Header Decoding	0.00003
Adaptation process	0.00015
PR Color-to-Infrared	75.02
PR Infrared-to-Color	75.10

Table 4. Adaptation times

75 ms has been measured both for color to infrared and infrared to color sensor switches. The Adaptation process time include all the states of the FSM of the Adaptation Controller excluding the partial reconfiguration time.

Partial bitstreams of the two Reconfigurable processing areas has not the same size. The color partial bitstream is about 5.8 MB whereas the infrared one is about 5.7 MB. As the two partial bitstreams has not the same size, the reconfiguration time differs lightly between the two adaptations.

A partial reconfiguration time of 75 ms is to long compared to usual times in Xilinx FPGA-based designs. This time represents about two image frames time in a 25 fps system. This unusual long time is due to the limitation of the partial reconfiguration in recent Altera FPGA. Actually, Altera FPGAs of the series V offers only an 1-Dimension column partial reconfiguration like the first Virtex-II Xilinx FPGAs [14]. However, two frames time is acceptable for sensor switch in non constrained applications. Future works will focus on implementation of this architecture in a recent Xilinx FPGA enabling partial reconfiguration such as series 7 FPGAs, to get better reconfiguration performance.

6 Conclusion

In this paper we have presented a self-adaptive architecture for multi-sensor vision system. The novelty of this work is that it proposes a new adaptive processing architecture which can deal with multiple and heterogeneous sensors. This architecture can dynamically adapt itself as a consequence of on-the-fly sensor switch. This self-adaptive architecture is based on a stream header concept and an adaptation controller to make the system aware of the sensor switch and hence to adapt the processing architecture.

This work offers a performance evaluation of the proposed concept in a FPGA implementation. The architecture was implemented in a Cyclone V Altera FPGA. Processing architecture is dynamically adapted by Dynamic and Partial Reconfiguration feature of FPGA. Rather than a performance comparison, the aim of this work is to give the proof of concept above all.

An average adaptation time of 75 ms has been measured. This time is mostly representative of the partial reconfiguration time. Because of the limitation of

the Altera's newly released partial reconfiguration technology, reconfiguration times are high. Nevertheless, this time remains admissible for non constrained applications. Resource overhead due to self-adaptation is almost insignificant compared to the resources utilization of the full design. The concept of this architecture is a promising start for further work on self-adaptive vision systems.

This work will be extended for multi-sensor vision systems with multiple and parallel streams, such as color-infrared image fusion systems. Future works will focus on improvement of the monitoring solution to enable a full self-awareness and environment-awareness to the vision system.

References

1. Junichi Nakamura. Image Sensors and Signal Processing for Digital Still Cameras. *Chapter 5, CMOS Image Sensors, pp. 143-178, CRC Press Taylor & Francis Group, August 2005.*
2. Marco Platzner, Jürgen Teich, Norbert Wehn. Dynamically Reconfigurable Systems, Architectures, Design Methods and Applications. *Chapter 18, pp. 375-415, Springer, 2010*
3. Merwan Birem and François Berry. DreamCam : A modular FPGA-Based Smart Camera Architecture. *Journal of Systems Architecture, Vol. 60, pp. 519 - 527, 2014.*
4. Tamás Raikovich, Béla Fehér. Application of partial reconfiguration of FPGAs in image processing *Conference on Ph.D. Research in Microelectronics and Electronics, 2010.*
5. Jalal Khalifat and Tughrul Arslan. A Novel Dynamic Partial Reconfiguration Design for Automatic White Balance. *NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2014.*
6. Jan van der Horst, Rien van Leeuwen, Harry Broers, Richard Kleihorst and Pieter Jonker. A Real-Time Stereo SmartCam, using FPGA, SIMD and VLIW. *In Proceedings of 2nd Workshop on Applications of Computer Vision, pp. 1-8, 2006*
7. Alessandro Muscoloni and Stefano Mattoccia. Real-Time Tracking with an Embedded 3D Camera with FPGA Processing. *International Conference on 3D Imaging (IC3D), December 2014.*
8. Texas Instruments. DaVinci Video Processors and Digital Media System-On-Chip. *tms320dm6446 Datasheet, 30 September 2010*
9. ON Semiconductor, formerly Aptina Imaging. 1/3-Inch CMOS Digital Image Sensor. *AR0130 Datasheet, 11/2014.*
10. E2V. 1.3 Mpixels B&W and Color CMOS image Sensor. *E2V Datasheet, 10/2011*
11. ULIS. Pico640 Gen2. *Ulis Datasheet, 01/2015*
12. Nicolas Ngan, Eva Dokladalova and Mohamed Akil. Dynamically Adaptable NoC Router Architecture for Multiple Pixel Streams Applications. *In IEEE International Symposium on Circuits and Systems (ISCAS'12), 2012.*
13. Altera. Cyclone V Device Overview. *CV-51001, 2015-06-12.*
14. S. Bhandari, S. Subbaraman, S. Pujari, F. Cancare, F. Bruschi, M. D. Santambrogio and P. R. Grassi. High Speed Dynamic Partial Reconguration for Real Time Multimedia Signal Processing. *15th Euromicro Conference on Digital System Design, 2012.*